In the claims:

1.   (Canceled)

2.   (Canceled)

3.   (Canceled)

4.   (Canceled)

5.   (Canceled)

6.   (Canceled)

7.   (Twice amended) A software program for facilitating the use of a distributed directory running in a computer network, the program comprising being stored on a recordable medium and including instructions for:

receiving a first event from an application;

converting the first event into XML data representing the first event;

transforming the XML data representing the first event to a first predetermined format by the transformation processor, the first predetermined format being responsive to the distributed directory;

transmitting the transformed XML data representing the first event to the distributed directory;

after receiving the first event from the application, receiving an first a second event from the distributed directory into an XML generator;

converting the first second event into XML data representing the second event;

transforming the XML data representing the second event to a first second predetermined format by athe transformation processor, the first second predetermined format being responsive to an application running in the computer network; and

transmitting the transformed XML data representing the second event transformed according to the first predetermined format to the application;

after receiving the first event from the application, receiving a third event from the distributed directory into an XML generator;

converting the third event into XML data representing the third event;

transforming the XML data <u>representing the third event</u> to a third predetermined format by the transformation processor, the third predetermined format being responsive to an application running in the computer network; and

transmitting the <u>transformed XML</u> data <u>representing the third event</u> ~~transformed according to the third predetermined format~~ to the application;

~~wherein the transformation processor includes an XSLT processor, the program further comprising instructions for:~~

providing a <u>first</u> stylesheet to ~~the~~ an XSLT processor, the stylesheet including formatting instructions for transforming XML data to the first predetermined format; and

providing a <u>second</u> ~~third~~ stylesheet to the XSLT processor, the <u>second</u> ~~third~~ stylesheet including formatting instructions for transforming XML data to the <u>second</u> ~~third~~ predetermined format wherein the first stylesheet is different from the <u>second</u> ~~third~~ stylesheet.

8.   (Previously canceled).

9.   (Twice amended) The software program of claim 7 further comprising instructions for:
receiving updates to the <u>first</u> stylesheet responsive to any changes in either the distributed directory or the application.

10.   (Amended) The software program of claim 7 wherein the transformed <u>XML</u> data <u>representing the second event</u> is transmitted to the application through an application shim to provide the transformed <u>XML</u> data <u>representing the second event</u> to the application by using a native application program interface for the application.

11.   (Amended) The software program of claim 7 further comprising instructions for:
detecting the <u>second</u> event through notification from an event handler of the distributed directory.

12.   (Canceled)

13.   (Twice amended) The software program of claim ~~12~~ <u>7</u> wherein the program further comprises instructions for:
providing a ~~second~~ <u>third</u> stylesheet to the XSLT processor, the ~~second~~ <u>third</u> stylesheet including formatting instructions for transforming XML data to the ~~second~~ <u>third</u> predetermined format.

14.   (Canceled)

15.   (Twice amended) A software program for facilitating the use of a distributed directory running in a computer network, the program comprising instructions for:

receiving an event from an application <u>prior to receiving event data from the distributed directory;</u>

<u>converting the event into markup language data;</u>

transforming the event to a predetermined format by a transformation processor, the predetermined format being responsive to the distributed directory; ~~and~~

<u>providing a transformation profile to the transformation processor, the transformation profile including formatting instructions for transforming the markup language data to the predetermined format;</u> and

transmitting the transformed event to the distributed directory.

16.   (Canceled)

17.   (Canceled)

18.   (Amended) A distributed computer system comprising:

a first processor connected to a network for executing computer code;

a second processor connected to the network for executing computer code;

a first memory connected to the first processor;

a second memory connected to the second processor;

a distributed directory, a portion of which being stored in the first memory;

an application, a portion of which being stored in the second memory;

<u>a first transformation profile for defining a first predetermined format for use by the distributed directory;</u>

a <u>second</u> transformation profile for defining a <u>second</u> predetermined format for use by the application;

software for detecting ~~an~~ <u>directory</u> event in the distributed directory;

<u>software for detecting an application event in the application prior to detecting the directory event;</u>

<u>software for transforming the application event to the second predetermined format by using a generic transformation tool and the first transformation profile; and</u>

<u>software for providing the transformed application event to the distributed directory;</u>

-4-

software for transforming the <u>directory</u> event to the predetermined format by using a generic transformation tool and the transformation profile; and

software for providing the transformed <u>directory</u> event to the application;

<u>whereby the distributed directory becomes aware of the application event by having the application event provided to the distributed directory in a transformed state and</u> whereby the application becomes aware of the <u>directory</u> event by having the <u>directory</u> event provided to the application in a transformed state.

19.   (Amended) The system of claim 18 further comprising:

software for converting the <u>directory</u> event to a generic data description before transforming the <u>directory</u> event.

20.   (Amended) The system of claim 18 further comprising:

an application shim for the application to receive the transformed <u>directory</u> event and provide the <u>directory</u> event to the application by using a native application program interface for the application.

21.   (Canceled)

22.   (Amended) The system of claim ~~21~~ 18 wherein the generic transformation tool utilizes a markup language and the software for transforming the event and the application event utilizes a transformation processor.